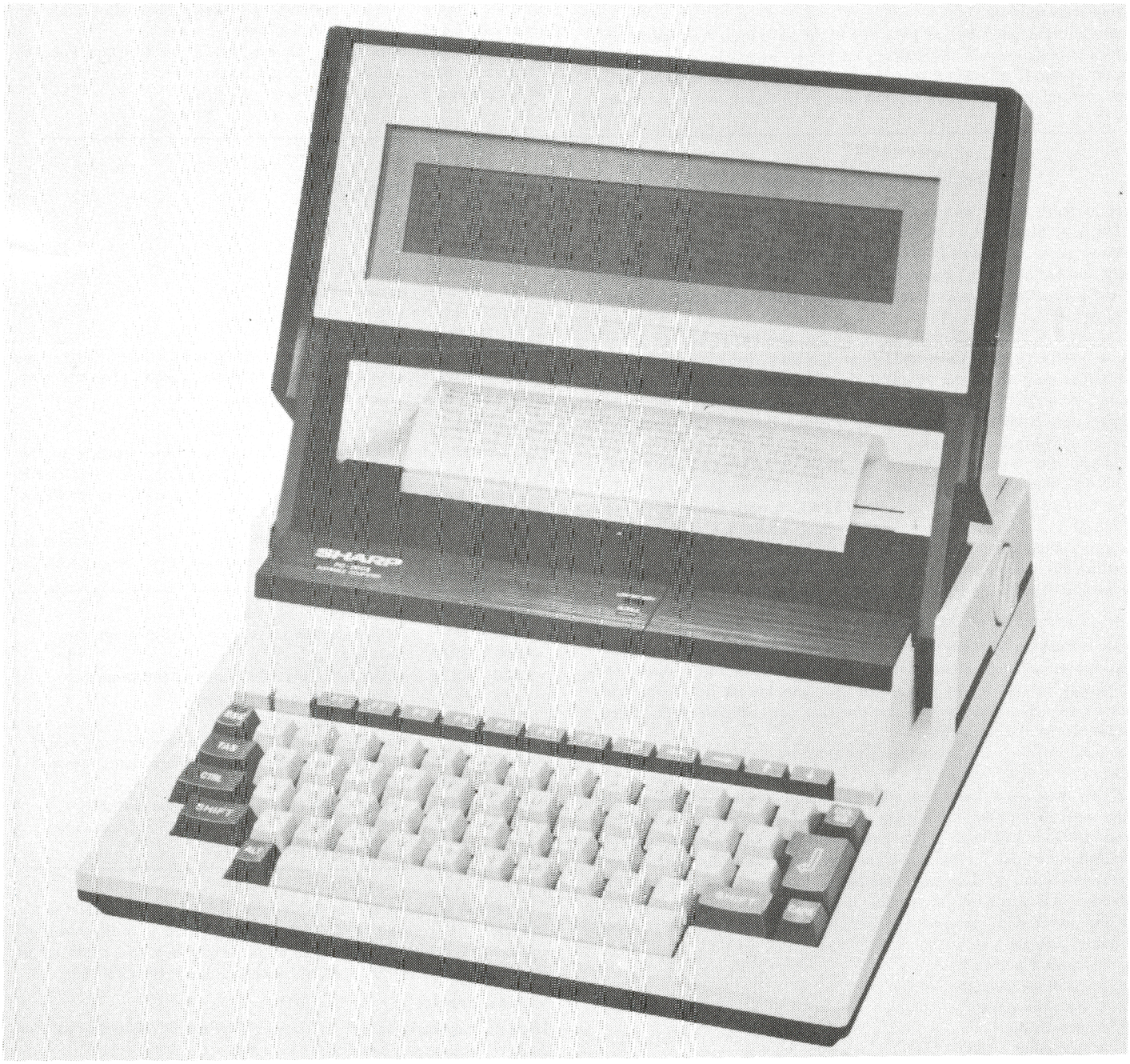
POCKET COMPUTER NEWSLETTER

© Copyright 1983 SCELBI Publications

Issue 26 — July

Photo *Sharp PC-5000 Portable Computer*



Another *Personal Information*™ product from SCELBI Publications.

SHARP STEPS UP IN SIZE

This fall Sharp Electronics Corporation will begin delivering a portable computer that it claims will combine the advantages of big computer capability with small computer convenience. The new machine, dubbed the PC-5000, weighs less than 11 pounds and can operate from a built-in rechargeable battery or from an a.c. adaptor. It is said to fit within an attache case.

It is supplied with 128K bytes of RAM memory, internally expandable to 256K (in 64K increments). Its typewriter-style keyboard is designed to facilitate word processing. Using an optional modem, the unit will function as a remote terminal.

The PC-5000 uses an 8088 microprocessor. In addition to the regular RAM memory, the unit has 128K of bubble memory storage. Another 128K of ROM in cartridges will be used to implement a disk operating system and BASIC.

The liquid crystal display folds over the keyboard when the unit is being transported. When open, the display presents up to eight lines of 80 characters of text or 640 by 80 dots (in excess of 51,000 pixels) for graphics in a bit-mapped mode. The display supports a full 256-character/symbol set.

A number of options may be added to the basic unit. A combination high-density dot-matrix thermal/impact printer inserts into the back of the PC-5000. It is claimed that this unit produces near letter quality copy. In the thermal mode it utilizes thermal paper. When a carbon ribbon is inserted, the printer will output on plain bond paper. The printer produces 80 characters per line at 12 characters per inch. The speed is approximately 37 characters per second. Optional 10 pitch formatting is also possible.

An optional 10-key modem/auto dialer can connect the computer to telephone lines through a standard Bell jack. The modem permits two-way conferencing through a loudspeaker provided on the face of the unit. Auto dialing of up to 10 different numbers can be accomplished using the manual keys. A communications software package can be used to access an essentially unlimited network. The modem allows up to 16 digits per number, has both ringer and receiver and can operate with touchtone or telepulse systems.

Double sided/double density disk drives, each providing up to 320K bytes of storage, will also be available. Alternately, programs can be loaded/saved on standard cassettes using a cassette interface.

Sharp promises initial software offerings including word processing, communications, spreadsheet analysis, executive planning and data base management which will be provided by arrangement with major software houses. Additional software compatability is said to be provided through the use of a MS-DOS system that will permit the use of existing MS-DOS software packages.

For additional information on the new Sharp PC-5000 Portable Computer, contact: *Sharp Electronics Corporation, Systems Division, 10 Sharp Plaza, P.O. Box 588, Paramus, NJ 07652.*

INSIDE THE SHARP PC-1250

Here is sleuth *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158,* with his report on the Sharp PC-1250:

Many users of Sharp's PC-1250 will have discovered by now that it has a number of BASIC statements and functions that are not covered by the instruction manual. A few of these were mentioned in Issue 24 of *PCN*.

The POKE, CALL, CSAVE M, CLOAD M and PEEK functions and the & prefix to indicate hexadecimal numerals operate in the same fashion as they do in the PC-1500. Statements in the RSV (Reserve) mode terminated by the @ symbol will also automatically execute as though they had been terminated by pressing the ENTER key when they are called into the display.

BASIC also contains COMS, ROM, INSTAT, OUTSTAT, SETCOM and KEY. These may appear in various valid statements such as SETCOM OFF, KEY OFF, KEY ON, OUTSTAT(numeric expression) and SETCOM, , , ,KI. These statements produce ERROR 8 ("device not present"). Does Sharp plan on making a modem for the PC-1250?

Although I have not been able to determine how they are used yet, the words ERROR and DEBUG are also tokenized by the 1250.

A Few Words on the CE-125 Printer

If you are trying to economize on paper, thermal rolls made for Texas Instruments calculators work fairly well. Rolls two inches in diameter can simply be laid in the paper chamber with the lid left open. The ones I have tried print in blue and they are not as legible as the ones furnished with the computer.

Keying SHIFT 6 will enter the symbol for the Japanese yen into the display.

A few additional characters may be obtained with the printer, but codes must be POKEd into a string variable to accomplish such output. Execute POKE &C698,&F5,1,0. This makes variable A a string variable containing the code 1. The zero byte terminates the string. If you now display A$, you will see some meaningless dots. But, LPRINT A$ will print the "not equal to" (≠) symbol. If the preceding POKE statement has the number 1 replaced by another value, other characters are printable, such as:

2 Greek sigma (Σ)
3 Boldface letter C
4 Boldface letter P
6 What is it?
7 And what is this?
9 Uppercase Greek pi (Π)
11 Hollow triangle
12 Solid triangle
14 Greek theta (θ)
15 Arrow pointing to the right
19 Character insertion marker
23 Symbol for yen
78 Underscore

Incidently, I hope that 1250 owners noticed the PRINT=LPRINT and PRINT=PRINT statements as described in the manual. (Too bad they did not incorporate that into the PC-1500!)

Memory Map

The accompanying table summarizes the accessible memory in the PC-1250. This accounts for 16K of ROM. It is believed that another 8K of ROM is located on the CPU chip and is not accessible to the PEEK directive.

Table *RAM Memory Map for PC-1250*

0000 – 3FFF	not used
4000 – 7FFF	ROM: operating system and BASIC interpreter
8000 – BFFF	not used
C000 – C7FF	RAM: including RSV, user's BASIC program, variables and system registers.
C800 – CFFF	not used in PC-1250
D000 – DFFF	duplicate of C000 – CFFF (where memory exists)
E000 – EFFF	duplicate of F000 – FFFF (where memory exists)
F000 – F800	not used
F800 – F8BF	system RAM (F8C0 – F8FF filled with FF)
FA00 – FAFF,	FB00 – FBFF . . . FF00 – FFFF are duplicate addresses for F800 – F8FF

A complete dump of the ROM from 4000 – 7FFF indicates the absence of the tables of BCD numbers required in calculations of transcendental functions. Thus, it must be assumed that these tables constitute part of the 8K of ROM believed to be on the CPU chip.

Machine Language

The CPU machine language opcodes used by the PC-1250 do not bear much resemblance to those of the PC-1500.

Table *ROM Memory Map for PC-1250*

4000 – 4020	JUMP addresses
4021 – 402F	Messages: BREAK, IN, ERROR
4030 – 4063	Addresses of tokenized words in initial-letter order
4064 – 4129	Addresses of tokenized words in token-code order
412A – 43DF	Table of BASIC words with tokens and ROM routine addresses
43E0 – 4414	Table of codes for shifted keys
4415 – 446B	Table for keys, used to convert row/column signals from keyboard into codes
446C – 4503	JUMP addresses
4504 – 45E2	Continuation of codes for character generation
45E3 – 7FFF	Appears to be all machine language code
C000 – C02F	Reserve memory area
C030	Stop byte (FF) marking beginning of user program area
C031 – C5CF	BASIC program area. Another FF byte is used as a end of program marker. Variables A(27) and above, as well as dimensioned arrays are also stored in this area. They begin at C5CF and work down towards the user's BASIC program. Each dimensioned variable has a 6-byte header that identifies it by name, type, length, dimensions and number of bytes per array element.
C5D0 – C5D7	Variable Z (or Z$)
C5D8 – D5DF	Variable Y (or Y$)
. . . . .	etc.
C698 – C69F	Variable A (or A$)
C6A0 – C70F	System RAM. Pointers in this area give the low order byte first. A number of pointers in this area are used by BASIC to keep track of program locations and locations of variables. Some of the more important ones are: C6E1 – C6E2 START OF BASIC C6E3 – C6E4 END OF BASIC C6FC – C6FD START OF VARIABLES There is no pointer to START OF EDITABLE (MERGED) PROGRAM. MERGE operates differently from in a PC-1500. C6E5 – C6E6 WAIT setting (low byte first) C6F8 – C6FA USING format settings C710 – C75F FOR/NEXT stack
C760 – C7AF	80-byte buffer used to form character strings
C7B0 – C7FF	80-byte buffer
F800 – F83B	Half of display memory
F83C – F83F	Mode settings. Bit 1 of F83C is Auto Print. DEGREE is set by bits 2 and 3 of F83C. GRAD by bits 2 of F83C and F83D. RADIAN by bit 2 of F83D. RUN, RRO and RSV are set by bits 0, 1 and 2 of F83E.
F840 – F87B	Other half of display memory
F87C – F87F	?
F880 – F8A7	Subroutine return address stack
F8A8 – F8AF	PASSword string
F8B0 – F8B7	Current value of random number
F8B8 – F8BF	Various settings. (F8BA – F8BB is the countdown timer storage area for the automatic 9-minute powerdown sequence.)

The information that follows is only a partial listing of the machine language instructions. The choices I have made for the names of the CPU registers are purely arbitrary since I have no technical information concerning the actual architecture of the CPU.

I have detected the presence of two flags, a zero flag (Z) and a carry flag (C). It is likely that other flags are also used.

The Accumulator

An 8-bit register, an accumulator, will be designated in my discussions by the letter A. The following list shows opcodes for operations that involve only this register:

OPCODE	MNEMONIC	OPERATION
02	LDA #nn	Accumulator loaded with immediate data nn
42	INA	A is incremented
43	DEA	A is decremented
58	RDA	Exchange digits of A
5A	RLA	Rotate bits to left through carry
D2	RRA	Rotate bits to right through carry
64	AND A #nn	Contents of A are ANDed with immediate data nn
65	OR A #nn	Contents of A are ORed with immediate data nn
66	BIT A #nn	Bit test. Flag Z is set or reset based on result of A being ANDed with nn
67	CPA #nn	Comparison. Flags C and Z set or reset based on the results of the subtraction of nn from A

Register W

This 16-bit register is used by some of the opcodes as a pointer to a location in memory. The symbol W in parentheses (W) will designate the contents of the location whose address is contained in register W. The high and low bytes of register W are designated WH and WL.

OPCODE	MNEMONIC	OPERATION
10	LDW #nnnn	W is loaded with the immediate value nnnn
11	LDWL #nn	WL is loaded with the immediate value nn
52	STA (W)	Contents of A stored into the location pointed to by W
57	LDA (W)	Contents of location pointed to by W loaded into A
D4	AND (W) #nn	Contents of (W) ANDed with the immediate value nn
D5	OR (W) #nn	Contents of (W) ORed with the immediate value nn
D6	BIT (W) #nn	Bit test. Flag Z set or reset based on result of (W) being ANDed with nn

Registers X and Y

Two more 16-bit registers, with high and low bytes designated as XH, XL, YH and YL are assumed to exist. The use of parentheses indicates the contents of the location pointed to by the X or Y registers.

OPCODE	MNEMONIC	OPERATION
03	LDXH #nn	Load XH with immediate value nn
06	INXL	Increment XL
07	DEXL	Decrement XL
C2	INXH	Increment XH
C3	DEXH	Decrement XH
26	STI A (X)	Store with increment. X is first incremented, then the contents of A are stored into (X)
27	STD A (X)	Store with decrement. X is first decremented, then the contents of A are stored into (X)
DA	EXA XH	Exchange A with XH
04	INY	Increment Y
05	DEY	Decrement Y
24	LDI A (Y)	Increment Y then load A with (Y)
25	LDD A (Y)	Decrement Y then load A with (Y)

Register S

The following discussion concerning the register arbitrarily designated S is somewhat speculative. However, repeated experiments have consis-

tently produced these results.

Register S appears to act as an 8-bit register that points to a stack located on the CPU chip. The effect of any opcode in the range from hexadecimal 80 to BF is to load S with a byte that causes S to point to one of the locations in that stack. I am arbitrarily calling these 1-byte addresses on the CPU chip 80 to BF. (They might actually, for instance, be 00 to 4F.) This stack does not appear to operate with PUSH or POP instructions.

OPCODE	MNEMONIC	OPERATION
80	LDS 80	S loaded with 80. (S) will designate contents of address 80
81	LDS 81	S loaded with 81. (S) will designate contents of address 81
. .	. . . . .	etc.
BF	LDS BF	S loaded with BF. (S) will designate contents of address BF

It appears that CPU registers A, X and Y are located in certain of the CPU stack locations as follows:

ADDRESS	REGISTER
82	A
83	XH
84	YL
85	YH
86	XL

Register W, however, is not located in this stack. Furthermore, subroutine return addresses do not appear to be stored here. Location 96 in this stack is used to save the TRACE mode setting!

The following instructions use (S), the contents of the address pointed to by register S:

OPCODE	MNEMONIC	OPERATION
55	STA (S)	Contents of A into (S)
59	LDA (S)	Contents of (S) into A
44	ADD (S) A	Contents of A added to (S) (without carry)
45	SUB (S) A	Contents of A subtracted from (S) (without carry)
46	AND (S) A	Contents of A and (S) are ANDed with the result left in (S)
47	OR (S) A	Contents of A and (S) are ORed with the result left in (S)
60	AND (S) #nn	(S) ANDed with immediate value nn
61	OR (S) #nn	(S) ORed with immediate value nn
62	BIT (S) #nn	Bit test. Flag Z set or reset as a result of (S) ANDed with nn
63	CP (S) #nn	Flags C and Z set or reset based on results of subtracting nn from (S)
C4	ADC (S) A	A added to (S) (with carry)
C5	SBC (S) A	A subtracted from (S) (with carry)
C6	BIT (S) A	Bit test. Z set or reset based on result of (S) AND A.
C7	CP (S) A	Flags C and Z set or reset based on subtraction of A from (S)
DB	EXA (S)	Exchange contents of A and (S)

One more instruction involving register S is:

OPCODE	MNEMONIC	OPERATION
30	STA S	A stored into S. (Values 00 to 7F have the same effect as 80 to FF.)

Flags

Flag C is set when the result of an addition produces a carry or when the result of a subtraction produces a borrow. It is cleared when no carry or borrow results. Note that in the case of subtraction, this is the opposite of the operation of the C flag in the PC-1500.

The Z flag is set whenever the result of an operation is zero. It is not affected by load and store instructions.

OPCODE	MNEMONIC	OPERATION
D0	SETC	Set flag C. (Flag Z is also set.)
D1	CLRC	Clear flag C. (Flag Z is set.)

Instructions for Branching and Subroutining

Jumps to absolute addresses are as follows:

OPCODE	MNEMONIC	OPERATION
79	JMP nnnn	Jump to address nnnn
7C	JZC nnnn	Jump if flag Z is clear
7E	JZS nnnn	Jump if flag Z is set
7D	JCC nnnn	Jump if flag C is clear
7F	JCS nnnn	Jump if flag C is set

Relative branch instructions exist for branching forward or backward through memory. The address reached by the branch is determined by counting from the final byte of the instruction, not from the initial byte of the following instruction.

OPCODE	MNEMONIC	OPERATION
2C	FWD nn	Forward branch nn bytes
28	FZC nn	Forward if flag Z is clear
38	FZS nn	Forward if flag Z is set
2A	FCC nn	Forward if flag C is clear
3A	FCS nn	Forward if flag C is set
2D	REV nn	Reverse branch nn bytes
29	RZC nn	Reverse if flag Z is clear
39	RZS nn	Reverse if flag Z is set
2B	RCC nn	Reverse if flag C is clear
3B	RCS nn	Reverse if flag C is set

Subroutine operations are controlled with the following directives:

OPCODE	MNEMONIC	OPERATION
78	JSR nnnn	Jump to subroutine at address nnnn
37	RTS	Return from subroutine

There Are More . . .

There are quite a few more opcodes whose operations have not been described in the preceeding lists. You are invited to pursue those on your own. The PC-1250 does not appear well-suited towards machine language programming for a variety of reasons. However, the above information may be sufficient to allow those interested to dabble in the subject and proceed on their own. Good luck!

Compatability of BASIC

It is claimed that programs written for the Sharp PC-1211 (and the equivalent Radio Shack PC-1) may be CLOADed directly into the PC-1250 without modification. Although this is generally true, there are a few exceptions.

A common technique used for conserving memory in the PC-1211 is to omit closing parentheses in expressions. The PC-1250, however, will not permit the use of this technique. The omitted parentheses will need to be inserted, if there is room! (Remember, though, the PC-1250 does have room for an additional 14 bytes over a PC-1211.)

Another possible incompatibility can cause problems when using a printer. Any programs for the PC-1211 that used the "wrap-around" based on a 16-character line in order to produce a particular printed format will need to be modified.

The PC-1250 uses the same unconventional FOR/NEXT looping as the PC-1211 (and early versions of the PC-1500). That is, the control variable is tested *before* it is incremented. The 1250 can also use implied multiplication.

Some Speed Comparisons

In an effort to see just how much faster the PC-1250 is than the 1211, I ran the following "benchmark" program:

```
10 A=1
20 FOR X=1 to 999
30 GOSUB 70
40 NEXT X
50 PRINT A
60 END
70 A=A+A/√(A+X)
80 RETURN
```

This program is an attempt to give a somewhat balanced comparison in that it contains a loop, a subroutine and some computation. The results I obtained were as follows:

PC-1211 693 seconds
PC-1250 218 seconds
PC-1500 107 seconds

It might be of interest to note that the same program took 57 seconds on a Commodore desktop microcomputer.

One reason the PC-1250 is slower than the PC-1500 is that the 1250 doesn't use link bytes (following line numbers) to point to the beginning of the next line of BASIC. It has to examine each byte of code to determine where one line ends and the next line begins.

In comparing available program space in two computers, it is not enough to just count the number of bytes. One computer may use more memory than another for tokens and for line numbers. To illustrate, the above program required 72 bytes in the PC-1500, but only 56 bytes in the PC-1211 and PC-1250. At that rate, the PC-1250's 1438 bytes can handle almost exactly the same size program as the (unexpanded) PC-1500 with its 1850 bytes!

Table *PC-1250 Character Codes and BASIC Token Values*

00	End of line	51	A	90	TO	C0	GRAD
		52	B	91	STEP	C1	PRINT
10	⌴ (insertion symbol)	53	C	92	THEN	C2	INPUT
11	Space	54	D	93	RANDOM	C3	RADIAN
12	"	55	E			C4	DEGREE
13	?	56	F	95	WAIT	C5	CLEAR
14	!	57	G	96	ERROR		
15	#	58	H			C9	CALL
16	%	59	I	99	KEY	CA	DIM
17	¥	5A	J			CB	DATA
18	$	5B	K	9B	SETCOM	CC	ON
19	π	5C	L			CD	OFF
1A	√	5D	M	9E	ROM	CE	POKE
1B	,	5E	N	9F	LPRINT	CF	READ
1C	;	5F	O				
1D	:			A0	SIN	D0	IF
1E	@	60	P	A1	COS	D1	FOR
1F	&	61	Q	A2	TAN	D2	LET
		62	R	A3	ASN	D3	REM
30	(	63	S	A4	ACS	D4	END
31	)	64	T	A5	ATN	D5	NEXT
32	>	65	U	A6	EXP	D6	STOP
33	<	66	V	A7	LN	D7	GOTO
34	=	67	W	A8	LOG	D8	GOSUB
35	+	68	X	A9	INT	D9	CHAIN
36	-	69	Y	AA	ABS	DA	PAUSE
37	*	6A	Z	AB	SGN	DB	BEEP
38	/			AC	DEG	DC	AREAD
39	^	7D	ASC	AD	DMS	DD	USING
		7E	VAL	AE	RND	DE	RETURN
40	ø	7F	LEN	AF	PEEK	DF	RESTORE
41	1						
42	2	81	AND	B0	RUN	E0	Line 0--
43	3	82	> =	B1	NEW	E1	Line 1--
44	4	83	< =	B2	MEM	E2	Line 2--
45	5	84	<>	B3	LIST	E3	Line 3--
46	6	85	OR	B4	CONT	E4	Line 4--
47	7	86	NOT	B5	DEBUG	E5	Line 5--
48	8	87	SQR	B6	CSAVE	E6	Line 6--
49	9	88	CHR$	B7	CLOAD	E7	Line 7--
4A	.	89	COM$	B8	MERGE	E8	Line 8--
4B	E	8A	INKEY$	B9	TRON	E9	Line 9--
		8B	STR$	BA	TROFF		
4E	_ (underscore)	8C	LEFT$	BB	PASS		
		8D	RIGHT$	BC	LLIST		
		8E	MID$	BD	PI		
				BE	OUTSTAT	FF	Stop byte
				BF	INSTAT		

THE CE-502A GENERAL STATISTICS MODULE

This plug-in ROM module for the Sharp PC-1500/Radio Shack PC-2 contains 16K of statistical programs, all in BASIC, located in memory in the addresses zero to &3FFF.

Since the module is plugged into the slot that is also used by the RAM expansion modules (such as the CE-151, CE-155 or CE-159), only the 2 kilobytes of unexpanded RAM is left available for user programs when the CE-502A is installed. A user program in RAM must begin with an executable label. This is because such a program is separated from the ROM module by the equivalent of a MERGE. It should also be noted that REServe memory is taken over by the CE-502A ROM module.

The module contains seven programs:

1. Means and Moments
2. t-Test, Paired
3. t-Test, Unpaired
4. One-Way ANOVA
5. Two-Way ANOVA
6. Contingency Table
7. Ranked Sum Test

Execution of any of these may be initiated by use of a REServe key. The correct REServe key can be conveniently identified by displaying the REServe memory template.

To get an idea of the amount of flexibility provided by the programs in this module, consider the options given the user in the Means and Moments program. Data may be entered either from the keyboard or from cassette. If the printer is connected, a choice is offered as to whether input data (and output) are to be printed. Inputs may be in the form of either grouped or ungrouped data. Editing of data is permitted. Data can be recorded on tape and a backup copy made. The second moment (variance) may be calculated with either n or n-1 used in the denominator.

The outputs provided by this program include the arithmetic, geometric and harmonic means, the second, third and fourth moments about the mean, the skewness and kurtosis and the number of entries made. However, it will not give you the standard deviation.

The t-tests for paired and unpaired data are used for determining the level of significance of the difference between measurements taken from two populations. The use of the test for differences between population means (the unpaired t-test) is based on the assumption that the two populations are normally distributed, with equal variances. Again, there are options for printing or not printing, entering data from the keyboard or cassette tape, inputting grouped or ungrouped data and recording data onto tape. The computer requests input of the hypothesized difference in means, with a default value of zero. The output includes the value of t, but the program does not calculate the level of significance. You will need to use tables for that. The number of degrees of freedom, together with the number of x entries and y entries, is also given.

The unpaired t-test (also known as the paired-difference test) requires only the assumption that the *differences* are normally distributed. The outputs here include the mean and standard deviation of the differences, the value of t and the number of degrees of freedom. Again, however, you will need to use a t-distribution table to interpret the results.

The programs for one-way and two-way analysis of variance (usually shortened to "ANOVA") provide for a variety of ways of handling the input data. Outputs include all necessary information, including the value of the F statistic. The use of an F-distribution table is required to interpret the results.

The two remaining programs involve nonparametric tests. The contingency table program gives the user the usual cassette options involving data. In this program there is a pleasant surprise. Not only is the value of chi-squared calculated, but its probability is determined too! This avoids the necessity of having to check a table to determine the level of significance. This is particularly useful in view of the fact that almost all chi-squared tables contain only certain selected values.

The final program, the Mann-Whitney Ranked Sum test, provides a way of comparing two means *without* the need for the assumptions required by the t-test. A Shell sort incorporated within the program is used to rearrange the input values, from each of the two samples, into numerical order. (It is possible to display or print the rankings after they have been sorted.) Outputs include the rank sum, rank mean and rank variance for both samples. The test statistics U (indicated as w) and z are given, but you will have to provide your own tables again. (Although the module contains a routine for determining approximate values of normal probability in lines 28205 – 28340, for some strange reason it is not used by *this* routine!)

There are many statistical tests in existence. The writers of the programs contained in this module had to make some choices. They wisely decided to omit some of the common tests that require a relatively small amount of computation, such as tests of hypotheses and determination of confidence intervals involving means, variances and proportions for single populations. Some users, however, might like to have seen inclusion of the chi-square test, linear regression and correlation. But, even 16K of memory can not hold everything, particularly when so many bells and whistles are included in the package.

One nice feature used in all of the programs in this module is a flashing of the default value in the display when yes or no inputs are requested. For example, the display of:

RECORD DATA (Y/N)?

is shown with the letter N flashing on and off. This tells the user that if ENTER is pressed without an input having been made, the computer will interpret the response as having been N for no! This feature is easy to get used to and it is convenient.

I would rate the manual as excellent. Although such a manual obviously can not include an entire course in statistics, the writers did take the trouble to include at least a brief explanation of what each of the tests is about. The manual ends with a list of examples of applications for which each of the tests in the module would be appropriate.

The manual documents the programs well. There are listings of the variables used by each program. In many cases the formulas used are also provided. This manual was obviously written in the U.S.A. It is not an atrocious translation from some other language!

I have not found any real bugs in the program, but there is a matter that comes pretty close to qualifying as such. The choice of algorithms used and the way in which they are implemented can produce some inaccurate results in certain circumstances.

The results frequently obtained for moments M3 and M4, as well as those for skewness and kurtosis, appear to be the worst. To illustrate: If the data values 865, 866 and 867 are entered, the following results are obtained:

RESULT	GIVEN BY PROGRAM	CORRECT VALUE
M2	1.00005333	1
M3	0.10000000	0
M4	-37.0370370	0.66666666
Skewness	0.09999200	0
Kurtosis	-37.0330867	0.66666666

A negative value for kurtosis is, of course, impossible. Note that the second moment also is not extremely accurate.

One cause of the errors is that during the accumulation of the sums ΣX^2, ΣX^3 and ΣX^4, the 10-digit capacity of the computer places limits on the accuracy of the sums. Then the required subtraction of nearly equal quantities having small relative errors produces a difference value having a large relative error.

About all you can do about this problem is code your data. Unfortunately, no provision appears to have been made for this in the program. (In the example presented, if you subtract 866 from each of the given input values and use the results as inputs, you will get correct values for M2, M3, M4, skewness and kurtosis.)

It seems to me that a better solution would have been for the programmer to have used another algorithm. The mean, $\bar{X}$, could have been calculated first. Next the numerator of M2 could have been calculated as $\Sigma(X-\bar{X})^2$ rather than as $\Sigma X^2 - (\Sigma X)^2/N$. (With this approach it would not have been possible to have accumulated the combined data from several tape files. There would have been ways to have gotten around that too, but it would have been rather complicated.) Similar approaches to the calculation of M3 and M4 would have eliminated the

significant errors noted in the illustration.

There is another inaccurate procedure used in these programs. The use of exponentiation is detrimental to the accuracy of the t-test and ANOVA programs as well as the Means and Moments program. Using, for example, X^2 in place of X*X has two bad effects: it is much slower and there is a loss of accuracy. The exponentiation operation uses logarithms to calculate an *approximate* result. This often contains slight inaccuracies. These inaccuracies are greatly magnified when subtraction of nearly equal quantities is required. That is exactly what happens in many statistical calculations. For instance, the error in M2 in the earlier example is entirely due to the use of X^2 in the program.

The programs could have been designed to operate somewhat faster. Part of the blame for a loss of speed has to go to the use of the exponentiation operation. The computer's response sometimes seems irritatingly slow when YES/NO inputs are required. While the sorting routine used in the Mann-Whitney test is quite efficient, a machine language sort would have really moved things along.

I have one more rather minor complaint. In four of the programs, data inputs are first placed into a string variable. Their numerical values are then later determined by use of the VAL function. Besides slowing down operation somewhat, this prevents one from using expressions such as 2/3, π or 968-750 as inputs. As a consequence, if you want to code input data, it is not nearly as easy to do. Their reason for using this approach was apparently to permit the input of E to signal the end of data. It would not have required a great deal of imagination to have achieved such a result another way.

However, my overall assessment of the CE-502A module is that it is a powerful and useful package for anyone doing much analysis of statistical data. Just be sure to take the skewness and kurtosis results with a grain of salt!

This review presented by: *Norlin Rober.*

Program *RPN Octal–Decimal–Hexadecimal Calculator*

```
100: CLS :CLEAR :A$
     ="0123456789AB
     CDEF":BA=16:BB
     =58:U=33:T$="&
     ":WAIT 0
110: ON ERROR GOTO
     730:GOTO 600
120: X$=""
130: K$=INKEY$ :IF
     K$=""THEN GOTO
     130
140: BEEP 1
150: K=ASC K$:IF (K
     >47AND K<BB)OR
     (K>64AND K<71
     AND BA=16)THEN
     LET X$=X$+K$:
     CLS :PRINT T$+
     X$:GOTO 130
160: IF X$=""THEN
     GOTO 240
170: IF X<>0THEN
     LET T=Z:Z=Y:Y=
     X
180: IF BA=10THEN
     LET X=VAL X$:
     GOTO 230
190: L=LEN X$:X=0:B
     T=1
200: FOR M=0TO L-1:
     O$=MID$ (X$,L-
     M,1):P=VAL O$
210: IF P=0THEN LET
     P=ASC O$-55:IF
     P<0THEN LET P=
     0
220: X=X+P*BT:BT=BT
     *BA:NEXT M
230: PRINT T$+X$,X:
     GOTO 250
240: IF K=13THEN
     LET T=Z:Z=Y:Y=
     X:GOTO 120
250: IF K=13AND X=0
     THEN LET T=Z:Z
     =Y:Y=X:GOTO 12
     0
260: IF K=43THEN
     LET X=Y+X:Y=Z:
     Z=T:GOTO 610
270: IF K=45THEN
     LET X=Y-X:Y=Z:
     Z=T:GOTO 610
280: IF K=42THEN
     LET X=Y*X:Y=Z:
     Z=T:GOTO 610
290: IF K=47THEN
     LET X=Y/X:Y=Z:
     Z=T:GOTO 610
300: IF K=24THEN
     LET X$="":X=0:
     PRINT T$+X$,X:
     GOTO 120
310: IF K=09THEN
     LET K=X:X=Y:Y=
     K:GOTO 610
320: IF K=10THEN
     LET K=X:X=Y:Y=
     Z:Z=T:T=K:GOTO
     610
330: IF K=11THEN
     LET K=T:T=Z:Z=
     Y:Y=X:X=K:GOTO
     610
340: IF K=02THEN
     LET X=-X:GOTO
     610
350: IF K=15THEN
     END
360: IF K=31THEN
     GOTO 470
370: IF K=01THEN
     GOTO 500
380: IF K=78THEN
     LET X=-X-1:
     GOTO 610
390: IF K=22THEN
     GOTO 550
400: IF K=79THEN
     GOTO 570
410: IF K=83THEN
     LET S=X:GOTO 6
     10
420: IF K=25THEN
     LET T=Z:Z=Y:Y=
     X:X=S:GOTO 610
430: IF K=61AND X<0
     AND BA<>10THEN
     LET N=UL+X:S$=
     "":GOTO 640
440: IF K=85THEN
     GOTO 590
450: IF X$=""THEN
     GOTO 610
460: PRINT T$+X$,X:
     GOTO 120
470: IF BA=16THEN
     LET BA=10:BB=5
     8:T$="%":PAUSE
     "DECIMAL":GOTO
     610
480: IF BA=10THEN
     LET BA=08:BB=5
     6:T$="#":PAUSE
     "OCTAL":GOTO 6
     10
490: IF BA=08THEN
     LET BA=16:BB=5
     8:T$="&":PAUSE
     "HEXADECIMAL":
     GOTO 610
500: K$=INKEY$ :IF
     K$=""THEN GOTO
     500
510: BEEP 1:K=ASC K
     $:IF K=11THEN
     LET X=√X:GOTO
     610
520: IF K=32THEN
     LET X=Y^X:Y=Z:
     Z=T:GOTO 610
530: IF K=24THEN
     LET X=0:Y=0:Z=
     0:T=0:X$="":
     PRINT T$+X$,X:
     GOTO 120
540: GOTO 120
550: GOSUB 670:X=(X
     2AND Y2)*B15*B
     15+(X3AND Y3)*
     B15+(X1AND Y1)
560: Y=Z:Z=T:GOTO 6
     10
570: GOSUB 670:X=(X
     2OR Y2)*B15*B1
     5+(X3OR Y3)*B1
     5+(X1OR Y1)
580: Y=Z:Z=T:GOTO 6
     10
590: PRINT "# BITS=
      ";U;" ";:
     INPUT U
600: BEEP 1:UL=1:
     FOR I=1TO U:UL
     =UL*2:NEXT I:
     CLS :PAUSE U;"
      BITS=";UL;" U
     P LIM":GOTO 61
     0
610: IF BA=10THEN
     PRINT T$,X:X$=
     "":GOTO 120
620: IF ABS X>UL
     THEN LET X=SGN
     X*UL
630: N=ABS X:S$="";
```

program listing
continued on next page

continuation of program listing from previous page

```
    IF X<0THEN LET
    S$="-"
640:X$=""
650:O=INT (N/BA):X
    $=MID$ (A$,N+1
    -BA*O,1)+X$:N=
    O:IF N<>0THEN
    GOTO 650
660:PRINT T$+S$+X$
    ,X:GOTO 120
670:B15=32768:IF
    ABS X>ULOR ABS
    Y>ULTHEN PRINT
    "TOO LARGE!":
    BEEP 2:GOTO 12
    0
680:IF X=0THEN LET
    X1=0:X2=0:X3=0
    :GOTO 700
690:X0=INT (X/B15)
    :X1=X-X0*B15:X
    2=INT (X0/B15)
    :X3=X0-X2*B15
700:IF Y=0THEN LET
    Y1=0:Y2=0:Y3=0
    :GOTO 720
710:Y0=INT (Y/B15)
    :Y1=Y-Y0*B15:Y
    2=INT (Y0/B15)
    :Y3=Y0-Y2*B15
720:RETURN
730:PAUSE "ERROR !
    !!":BEEP 3:
    PRINT "":GOTO
    120
STATUS 1
                    2014
```

RPN CALCULATOR FOR PROGRAMMERS

This program simulates a RPN hexadecimal, decimal and octal calculator on the PC-1500. It was provided by: *Michael H. Frey, 1536 Pine Street, Philadelphia, PA 19102.*

The program allows the following functions: addition, subtraction, multiplication, division, square root, Y to the X power, NOT, OR, AND, 2's complement and one memory with recall.

The upper limit for calculations is initially set at 8589934592 (33 bits for a word length) with an accuracy of +/- one digit in the last place.

The simulator has a four register stack labeled X, Y, Z and T. The functions of the program are summarized in the accompanying table. Numbers are entered as in a normal calculator. Note that the hexadecimal mode also uses the "digits" A through F. Illegal characters are treated as an ENTER. The program recognizes that A – F are valid in the hexadecimal mode and that 8, 9 and A – F are not valid in the octal mode. Two numbers are normally displayed at a time. The exceptions are when actually entering a number or when in the decimal mode, the left hand number is hexadecimal or octal and the right hand number is the decimal equivalent.

The various numerical modes are indicated by preceding the left hand number by a & for hexadecimal, a % for decimal and a # for octal. Inputs must be integer numbers and the left hand number is always displayed as the truncated integer of the right hand number.

Table *Functions of RPN Calculator for Programmers*

KEY SEQUENCE/ FUNCTION	REGISTER CONTENTS AFTER OPERATION				
	X	Y	Z	T	M=MEMORY
ENTER	X	X	Y	Z	M
CL	0	Y	Z	T	M
SHIFT CL	0	0	0	0	M
↕	Y	X	Z	T	M
↓	Y	Z	T	X	M
↑	T	X	Y	Z	M
S=STORE	X	Y	Z	T	X
RCL	M	X	Y	Z	M
+–*/=[]	Y[]X	Z	T	T	M
SML=change sign	–X	Y	Z	T	M
SHIFT √	√X	Y	Z	T	M
SHIFT ^	Y^X	Z	T	T	M
N=NOT	NOT X	Y	Z	T	M
O=OR	X OR Y	Z	T	T	M
&=AND	X AND Y	Z	T	T	M
=	x(2'c) displays 2's complement of x if x is negative	Y	Z	T	M
MODE	hexadecimal->decimal->octal->hexadecimal->...				
U	SET NUMBER OF BITS				
OFF	RETURNS TO PC-1500 RUN MODE				

The POCKET COMPUTER NEWSLETTER is Available!

By Subscription Only: for a calendar year period (January – December). You get all issues published to date for the calendar year in which you subscribe, at the time you subscribe.

MC/VISA Phone Subscriptions: (203) 888-1946

☐ 1982 Regular Subscriber (Issues 11 – 20). $30.00 in U.S. (U.S. $36.00 to Canada. U.S. $45.00 elsewhere.)

☐ 1982/83 Subscriber (Issues 11 – 30). $60.00 in U.S. (U.S. $72.00 to Canada. U.S. $90.00 elsewhere.)

☐ 1983 Regular Subscriber (Issues 21 – 30). $36.00 in U.S. (U.S. $42.00 to Canada. U.S. $50.00 elsewhere.)

Orders must be accompanied by payment in full. We do not issue invoices for the POCKET COMPUTER NEWSLETTER. Thank you for your remittance.

Name:_____

Addr: _____

City: _____ State: ___ Zip: _____

MC/VISA #: _____ Expires: _____

Signature: _____

P.O. Box 232, Seymour, CT 06483

FROM THE WATCH POCKET

PC-1/PC-1211 owners with an interest in astronomy might want to write to *Fred Klein, 12225 Magdalena, Los Altos, CA 94022*, for information about a collection of programs he is self-publishing. The 100 page (8-1/2 by 11 inches) publication is described as "a collection of 13 programs for observers, astrophotographers and Dobsonians." Mr. Klein, holder of a Ph.D., appears to have written an astronomical works that will be of interest to serious astronomical enthusiasts.

Speaking of serious works, **Inframetrics, Inc.**, *25 Wiggins Ave., Bedford, MA 01730*, has announced a "radiometric computer" that performs "non-contact temperature measurement calculations in less than one second." The unit appears to be a Quasar HHC packaged with special software and a custom calibration module. Write to the firm for pricing, (about $650.00 for the basic package), literature and additional technical details.

Prices are starting to drop on add-on memory modules for the PC-1500/PC-2. Radio Shack is selling their 4K modules for $39.00. **Atlantic N.E. Marketing**, *P.O. Box 921, Marblehead, MA 01945*, telephone (617) 639-0285, says they have a limited supply of CE-151 4K modules available at $25.00! Word is that the talked-about 16K RAM module coming this fall (September) from Sharp will be designated the CE-161, will have a list price of $170.00 and will be battery-backed. (The 8K battery-backed CE-159 has been dropped to $110.00.)

PCN is currently published 10 times a year. There will not be any August issue! Have a pleasant summer. — *Nat Wadsworth, Editor*